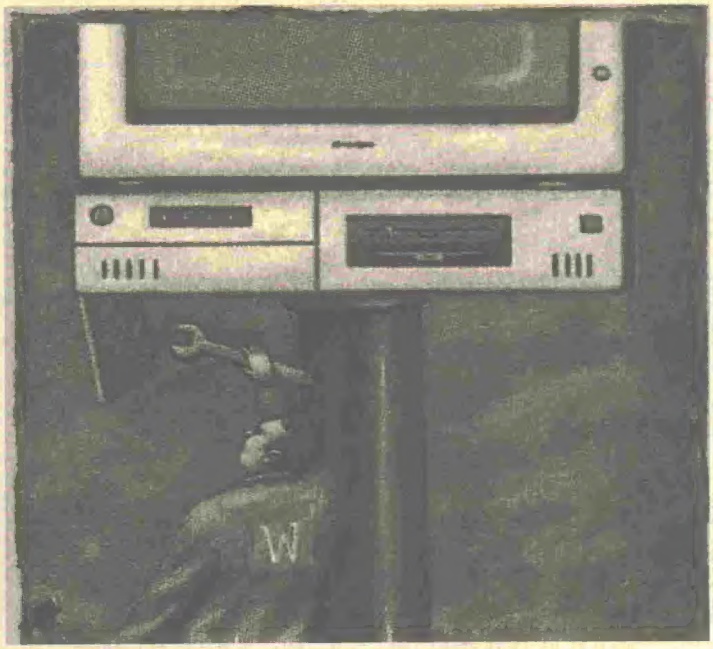
ขจร โรจนเมธินทร์

Visual BASIC : มุมมองใหม่ของ Data Access



นำเสนอมุมมองของดาต้าออปเจ็กต์ ในแง่มุมใหม่ และสามารถนำไปประยุกต์ใช้งานต่อได้เป็นอย่างดี

สาเหตุที่ผู้เขียนตั้งชื่อบทความว่ามุมมองใหม่ของ Data Access ก็เพราะผู้เขียนตั้งใจจะนำเสนอแนวคิดเกี่ยวกับการเรียกใช้ดาต้าเบสในวิชวลเบสิกแบบใหม่ ก่อนหน้านี้ผู้เขียนได้เคยกล่าวถึงการใช้ดาต้าเบสโดยไม่ใช้ดาต้าคอนโทรลมาแล้วในฉบับที่ 124 ซึ่งผู้อ่านคงจะเห็นถึงวิธีการใช้ดาต้าออปเจ็กต์แบบคร่าว ๆ ไปแล้ว ในบทความนี้ผู้เขียนจะยังกล่าวถึงการใช้ดาต้าออปเจ็กต์อีก แต่จะเปลี่ยนมุมมองในการมาดาต้าออปเจ็กต์ให้เข้าใจง่ายขึ้น

มุมมองของดาต้าออปเจ็กต์

ตามที่ผู้เขียนเคยกล่าวถึงออปเจ็กต์ Database ว่าเป็นกลุ่มของข้อมูลที่มีความสัมพันธ์กันในหัวข้อใดหัวข้อหนึ่ง ส่วนประกอบของดาต้าออปเจ็กต์อาจเป็นตารางข้อมูล คิวรี่ อินเด็กซ์ ในบทความนี้ผู้เขียนจะเน้นที่ตารางข้อมูล (Table) ก่อน ในดาต้าเบสแต่ละอันอาจประกอบด้วยตารางข้อมูลหลาย ๆ ตารางเก็บอยู่ด้วยกัน ตารางแต่ละอันสามารถแบ่งออกเป็น 2 ส่วน คือ ส่วนที่เป็นข้อมูลจริง ๆ ที่เก็บในตาราง เราสามารถเรียกใช้ส่วนนี้ได้จากออปเจ็กต์ Table และส่วนที่เป็นโครงสร้างของตาราง เช่น จำนวนฟิลด์ข้อมูลในตาราง ชื่อฟิลด์ ชนิดข้อมูลของฟิลด์ ความยาวของฟิลด์ เป็นต้น ในส่วนนี้จะเรียกใช้ได้จากออปเจ็กต์ TableDef สำหรับออปเจ็กต์ทั้งสองอันนี้มีความสัมพันธ์กับออปเจ็กต์ Database เป็นอย่างมาก โดยก่อนที่จะใช้งานได้ เราจะต้องกำหนดออปเจ็กต์ Database ขึ้นมาก่อน หลังจากที่กำหนดดาต้าเบสขึ้นมาแล้ว นอกจากนี้แล้วยังมีออปเจ็กต์ Field, Index, QueryDef, และ Snapshot ซึ่งเกี่ยวข้องกับออปเจ็กต์ Database ด้วย แต่ผู้เขียนคงกล่าวถึงเฉพาะ ออปเจ็กต์ Field ก่อนเท่านั้น

พรอพเพอร์ตี้ส์ของออปเจ็กต์ Database

CollatingOrder

พรอพเพอร์ตี้ส์นี้สามารถใช้ได้กับทั้งดาต้าออปเจ็กต์และเรคอร์ดเซต หมายถึงกลุ่มของเรคอร์ดซึ่งก็คือออปเจ็กต์ Table นั่นเอง พรอพเพอร์ตี้ส์จะเก็บค่าที่แสดงถึงวิธีการเรียงลำดับของข้อมูลในดาต้าเบสหรือเทเบิล ถ้าผู้ใช้ต้องการกำหนดวิธีการเรียงลำดับข้อมูลเองจะต้องใช้ประโยค Compact Database ช่วย โดยกำหนดที่แฟลก locale ของประโยค

Connect

เป็นพรอพเพอร์ตี้ส์ซึ่งจะใช้กำหนดหรืออ่านค่าวิธีที่ตัว Access Engine จะใช้ติดต่อกับดาต้าเบสที่สร้างจากโปรแกรมอื่นที่ไม่ใช่ Access ในการใช้งานจริงผู้ใช้ต้องสร้างไฟล์ .INI ที่มีชื่อเดียวกับแอพพลิเคชันที่เราเขียนขึ้น โดยไฟล์นี้จะต้องเก็บไว้ในไดเรคทอรีของวินโดวส์ จะใช้เก็บค่าที่จะควบคุมการทำงาน เช่น DLLs ที่ใช้, ค่า Time out, ขนาด Page frame เป็นต้น (ดูตัวอย่างที่ 1)

ตัวอย่างที่ 1

รูปแบบ : DatabaseObj.Connect [= setting$]
DatabaseObj.TableDefs(Index%|Table$).Connect [= setting$]

DatabaseObj	หมายถึง ชื่อดาต้าออปเจ็กต์ที่เรากำหนดขึ้นด้วยประโยค Set
setting$	อาจกำหนดเป็น "(none)" หมายถึง MSAccess, "dBASE III" "dBASE IV", "Paradox;", "Btrieve;", "FoxPro 2:0", "FoxPro 2.5", "ODBC;"
Index%	หมายถึง อินเด็กซ์ที่ใช้อ้างถึงตารางในดาต้าเบส
Table$	หมายถึง ชื่อตารางซึ่งเราต้องการกำหนดวิธีการ Connect

ตัวอย่างที่ 2

รูปแบบ : DatabaseObj.OpenTable(TableName$[, option%])

TableName$	เป็นชื่อตารางที่ต้องการเปิด
option%	เป็นวิธีการที่จะเปิดตาราง สามารถกำหนดเป็น 1 (ไม่ยอมให้ผู้ใช้อื่นแก้ไขตาราง ในระบบมัลติยูสเซอร์), 2 (ไม่ยอมให้ผู้ใช้อื่นอ่านตาราง ในระบบมัลติยูสเซอร์), 4 (อ่านข้อมูลจากตารางได้เท่านั้น)

ตัวอย่าง :
```
Dim MyTable As Table
Dim MyDb As Database
Set MyDb = OpenDatabase("d:\vb3\biblio.mdb")
Set MyTable = MyDb.OpenTable("Authors")
```

TableDefs Collection

พรอพเพอร์ตี้ส์มีความสำคัญมาก เพราะเราสามารถใช้พรอพเพอร์ตี้ส์นี้เพื่ออ้างถึงคอลเลคชัน TableDefs ซึ่งก็คือกลุ่มของออปเจ็กต์ TableDef และออปเจ็กต์ TableDef ได้ทันทีโดยที่ไม่ต้องกำหนดออปเจ็กต์ TabelDef ขึ้นด้วยประโยค Set

```
DatabaseObj.TableDefs.Count
```

รูปแบบนี้เป็นการใช้พรอพเพอร์ตี้ส์ TableDefs เพื่ออ้างถึงคอลเลคชัน TableDefs ซึ่งมีพรอพเพอร์ตี้ส์ให้ใช้ตัวเดียว คือ Count ที่เก็บค่าจำนวนเทเบิลที่มีในคอลเลคชัน TableDefs ดังนั้นเราจึงทราบว่ามีเทเบิลกี่เทเบิลในดาต้าเบสนี้

```
DatabaseObj.TableDefs(Index%|Table$).Property [=setting]
```

รูปแบบนี้เป็นการใช้พรอพเพอร์ตี้ส์ TableDefs เพื่ออ้างถึงออปเจ็กต์ TableDef โดยเราต้องระบุเทเบิลที่จะใช้ด้วยการส่งค่าอินเด็กซ์ที่อ้างถึงเทเบิลหรือชื่อเทเบิลที่จะใช้ไป ทำให้เราสามารถเข้าถึงพรอพเพอร์ตี้ส์ของออปเจ็กต์ TableDef ได้ทันที ด้วยวิธีการนี้เราจะสามารถเข้าถึงข้อมูลที่เป็นโครงสร้างของตาราง เช่น ชื่อเทเบิล

ตัวอย่างที่ 3

รูปแบบ : CreateDatabase(dbname$[, lang%[, option%]])

dbname$	ใช้กำหนดชื่อดาต้าเบสไฟล์ที่ต้องการสร้าง
lang%	ใช้กำหนดชนิดภาษาข้อมูลที่จะเก็บในดาต้าเบส
option%	ใช้กำหนดให้เข้ารหัสข้อมูล หรือสร้างในฟอร์แมต Access 1.0

ตัวอย่าง :
```
Dim NewDb As Database
Set NewDb = CreateDatabase("d:\newdata.mdb",
DB_LANG_GENERAL, DB_ENCRYPT)
```

เมธอดของออปเจ็กต์ Database

Close

เมธอดนี้ใช้ปิดดาต้าเบสที่เปิดใช้งานอยู่ในขณะนั้น โดยการปิดดาต้าเบสนี้จะปิดเทเบิลหรือเรคอร์ดเซตทั้งหมดที่เปิดใช้งานอยู่ขณะนั้นทันที

OpenTable

เมธอดนี้ใช้เปิดเทเบิลขึ้นมาใช้งาน โดยมากจะต้องใช้ร่วมกับประโยค Set เสมอ เพื่อกำหนดค่าให้กับออปเจ็กต์ Table ต่อ (ดูตัวอย่างที่ 2)

ฟังก์ชันของออปเจ็กต์ Database

CreateDatabase

เป็นฟังก์ชันที่ใช้สร้างดาต้าเบสใหม่ โดยผู้ใช้ต้องกำหนดชื่อไฟล์ที่จะสร้างและส่งตัวเลือกต่าง ๆ ไปให้ ต้องใช้ร่วมกับประโยค Set เสมอ (ดูตัวอย่างที่ 3)

OpenDatabase

ฟังก์ชันนี้ใช้สำหรับเปิดดาต้าเบสขึ้นมาใช้งาน โดยระบุชื่อไฟล์ดาต้าเบสที่มีอยู่แล้วส่งให้ฟังก์ชัน การใช้งานต้องใช้ร่วมกับประโยค Set

พรอพเพอร์ตี้ส์ของออปเจ็กต์TableDef

Fields Collection

พรอพเพอร์ตี้ส์ Fields นี้ใช้สำหรับอ้างถึงคอลเลคชัน Fields ซึ่งก็หมายถึง กลุ่มของออปเจ็กต์ Field ได้โดยตรง แทนที่จะต้องกำหนดเป็นออปเจ็กต์อย่างชัดเจน การใช้งานจะคล้ายกับพรอพเพอร์ตี้ส์ TableDefs Collection การใช้งานแบ่งออกเป็น 2 ประเภท คือ

```
TbDefName.Fields.Count
```

การใช้ในรูปแบบนี้เป็นการอ้างถึงกลุ่มของออปเจ็กต์ Field คอลเลคชันของ Field นี้มีพรอพเพอร์ตี้ส์ให้ใช้คือ Count ซึ่งเก็บจำนวนออปเจ็กต์ Field ที่มีในกลุ่มนั้น

```
TbDefName.Fields(Index%).Property [= setting]
```

การใช้งานแบบนี้เป็นการอ้างถึงออปเจ็กต์ Filed แต่ละอันโดยตรง ผู้ใช้ไม่ต้องกำหนดออปเจ็กต์ขึ้นเอง สำหรับฟิลด์ที่ต้องการอ้างถึงจะกำหนดที่ Index% ส่วนจะใช้พรอพเพอร์ตี้ส์ใดก็เรียกต่อท้ายได้ทันที

ตัวอย่าง แสดงการใช้คอลเลคชัน Fields ของออปเจ็กต์ MyTableDef โดยเราจะเรียกใช้พรอพเพอร์ตี้ส์ Name ของฟิลด์ที่ 1

```
Dim MyTableDef As TableDef
Set MyTableDef = MyDB.TableDefs(1)
Print MyTableDef.Fields(1).Name
```

Name

พรอพเพอร์ตี้ส์นี้จะเก็บชื่อตารางหรือเทเบิลทั้งหมดที่มีอยู่ในดาต้าเบส โดยอาจเลือกใช้กับออปเจ็กต์ TableDef โดยตรง หรือใช้กับคอลเลคชัน TableDefs แทนก็ได้

พรอพเพอร์ตี้ส์ของออปเจ็กต์ Table

BOF

พรอพเพอร์ตี้ส์ BOF นี้เป็นตัวชี้ว่าเรคอร์ดปัจจุบันขณะนั้นเป็นเรคอร์ดแรกของเทเบิลหรือไม่ ถ้าเรคอร์ดปัจจุบันอยู่ที่ต้นไฟล์ ค่า BOF จะเป็น True มิฉะนั้นจะเป็น False ปกติแล้วควรใช้พรอพเพอร์ตี้ส์นี้ตรวจสอบหลังจากการใช้เมธอด Move ทุกครั้งเพื่อมิให้เกิดการพยายามเรียกใช้เรคอร์ดที่อยู่นอกเหนือจากไฟล์

EOF

พรอพเพอร์ตี้ส์ EOF จะคล้ายกับ BOF แต่ความหมายตรงกันข้าม คือ จะเป็นตัวชี้ว่าเรคอร์ดปัจจุบันอยู่ที่ท้ายไฟล์แล้ว โดยมากเราจะใช้พรอพเพอร์ตี้ส์นี้กับ Do...Loop และ MoveNext เพื่อเรียกเรคอร์ดทั้งหมดในตารางขึ้นมาใช้งาน ซึ่งเมื่อใช้ MoveNext กับเรคอร์ดสุดท้ายแล้ว ค่า EOF จะเป็น True

Fields Collection

การใช้งานพรอพเพอร์ตี้ส์ Fields collection ของออปเจ็กต์ Table นี้จะคล้ายกับของออปเจ็กต์ TableDef เราสามารถใช้พรอพเพอร์ตี้ส์นี้เพื่ออ้างถึงกลุ่มของฟิลด์และอ้างถึงออปเจ็กต์ Fields ได้ทันที แต่ในกรณีที่ใช้กับออปเจ็กต์ Table นี้ เราจะสามารถใช้พรอพเพอร์ตี้ส์ Value ของออปเจ็กต์ Fields ได้ ถ้าเป็นออปเจ็กต์ TableDef จะอ้างถึงพรอเพอร์ตี้ส์ Value ของ Fields ไม่ได้ ดังนั้นในกรณีนี้ส่วนใหญ่จึงใช้อ่านค่าข้อมูลที่เก็บอยู่ในแต่ละฟิลด์ขึ้นมาใช้ เช่น เมื่อต้องการค่าของฟิลด์ที่ 1 ของตารางก็จะใช้รูปแบบดังนี้

```
MyTable.Fields(1).Value
```

NoMatch

พรอพเพอร์ตี้ส์ NoMatch นี้โดยมากจะใช้ร่วมกับเมธอด Find ในการค้นหาค่าข้อมูลที่เก็บในตาราง โดยเราจะตรวจสอบค่าของพรอพเพอร์ตี้ส์นี้เพื่อดูว่าการค้นหาทำได้สำเร็จหรือไม่ ในกรณีที่ค้นหาข้อมูลที่ต้องการพบค่า NoMatch ก็จะเป็น False แต่ถ้าค้นหาไม่พบค่าก็จะเป็น True

เมธอดของออปเจ็กต์ Table

AddItem

เมธอดนี้จะใช้สำหรับเพิ่มเรคอร์ดที่อยู่ในบัฟเฟอร์ไว้ที่ท้ายตาราง หลังจากนั้นจะเซตค่าของแต่ละฟิลด์ในบัฟเฟอร์ให้เป็น Null แต่ในความเป็นจริงแล้วข้อมูลจะยังไม่ได้ถูกเขียนไว้ในตารางข้อมูลจนกว่าผู้ใช้จะเรียกเมธอด Update หรือใช้เมธอด Move เท่านั้น เรคอร์ดใหม่ที่เพิ่มเข้าไปนี้ยังไม่ได้เรียงลำดับจนกว่าจะใช้เมธอด Refresh หรือปิดดาต้าเบสแล้วเปิดใหม่เท่านั้น

Close

การใช้เมธอดนี้กับออปเจ็กต์ Table จะเป็นการปิดเฉพาะเทเบิลที่ใช้ด้วยเท่านั้น แต่ถ้าใช้เมธอด Close กับดาต้าออปเจ็กต์ ก็จะเป็นการปิดเทเบิลทั้งหมดในดาต้าเบสนั้น

Delete

เมธอด Delete ใช้สำหรับลบเรคอร์ดปัจจุบัน โดยทุกฟิลด์จะถูกเซตค่าให้เป็น Null เรคอร์ดที่ลบไปแล้วไม่สามารถอ้างถึงได้อีก ดังนั้นหลังจากลบไปแล้วก็ควรเรียกเมธอด Move เพื่อเลื่อนเรคอร์ดปัจจุบันไปที่เรคอร์ดอื่น มิฉะนั้นจะเกิด Error ขึ้นได้ นอกจากนี้ก่อนจะใช้เมธอด Delete ก็ควรจะแน่ใจว่าขณะนั้นไม่ได้อยู่ที่ตำแหน่ง BOF หรือ EOF ของไฟล์ ในกรณีที่ต้องการลบข้อมูลมาก ๆ ควรใช้คำสั่ง SQL ซึ่งจะทำงานได้เร็วกว่า

FindFirst, FindLast, FindNext,FindPrevious

เมธอดนี้ใช้สำหรับค้นหาเรคอร์ดที่ต้องการจากเทเบิล โดยเราต้องกำหนดค่าที่ต้องการให้ค้นหาไว้เป็นเงื่อนไขหรือ Criteria การใช้งานเมธอด Find นี้จะต้องใช้ร่วมกับการตรวจสอบค่าพรอพเพอร์ตี้ส์ NoMatch การค้นหาข้อมูลโดยใช้เมธอด Find นี้จะช่วยให้การทำงานเร็วขึ้น เพราะแทนที่ผู้ใช้จะต้องตรวจสอบค่าเองทั้งหมด ก็สามารถเลือกได้เฉพาะเรคอร์ดที่ต้องการเท่านั้น

MoveFirst, MoveLast, MoveNext, MovePrevious

เมธอด Move จะใช้สำหรับเลื่อนเรคอร์ดปัจจุบันไปยังตำแหน่งที่ต้องการ โดยมากเรามักใช้ MoveNext มาประกอบในการวนลูปไปยังเรคอร์ดทั้งหมดที่มีในเทเบิล

Update

การใช้เมธอด Update ก็เพื่อจะเซฟการเปลี่ยนแปลงที่เกิดกับข้อมูล ไม่ว่าจะเป็นเรคอร์ดที่เพิ่มใหม่หรือกรณีที่มีการแก้ไขค่าในฟิลด์ใดฟิลด์หนึ่ง

พรอพเพอร์ตี้ส์ของออปเจ็กต์ Field

Name, Size, Type

การใช้พรอพเพอร์ตี้ส์ Name, Size, Type ของออปเจ็กต์ Fields นี้ หากนำไปใช้ประกอบกับออปเจ็กต์ Table หรือ TableDef ก็จะให้ค่าเหมือนกันคือ ใช้อ่านชื่อ ขนาด ชนิดข้อมูลของฟิลด์ ขึ้นมาใช้

Value

พรอพเพอร์ตี้ส์ Value ของออปเจ็กต์ Fields จะใช้ได้กับออปเจ็กต์ Table เท่านั้น ไม่สามารถใช้กับออปเจ็กต์ TableDef ได้ การใช้งานก็เพื่ออ่านค่าข้อมูลที่เก็บอยู่ในฟิลด์นั้น

เมธอดของออปเจ็กต์ Field

Append

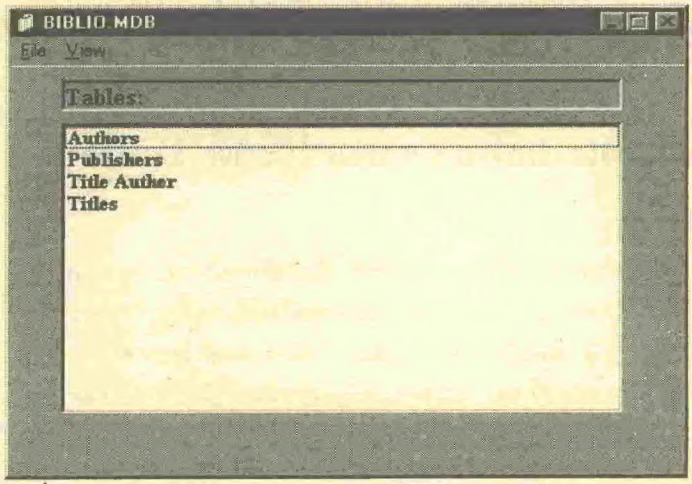
เมธอด Append นี้สามารถใช้เพิ่มออปเจ็กต์ TableDef, Index, Field ใหม่เข้าในคอลเลคชันของ TableDefs, Indexes, Fields ดังนั้นจึงสามารถใช้เมธอด Append นี้สำหรับสร้างตารางใหม่ในดาต้าเบส เพิ่มฟิลด์และอินเด็กซ์ใหม่ในตารางได้

โปรเจ็กต์ Database Explorer

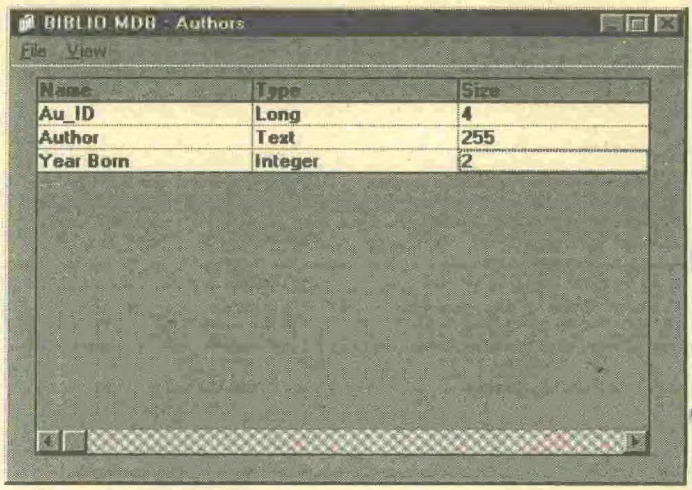
โปรแกรมตัวอย่างที่ผู้เขียนใช้แสดงการใช้ดาต้าออปเจ็กต์ในบทความนี้ คือ Database Explorer ผู้เขียนต้องการให้ใช้โปรแกรมนี้สำหรับเรียกดูโครงสร้างของดาต้าเบสว่าเก็บตารางใดไว้บ้าง สามารถเปิดใด ๆ ในดาต้าเบสขึ้นมาเพื่อเรียกดูโครงสร้างของตารางและข้อมูลในตารางได้ แต่สำหรับโครงสร้างของตารางนั้น Database Explorer จะไม่แสดงในส่วนอินเด็กซ์ของตาราง ซึ่งความจริงวิธีการเรียกส่วนอินเด็กซ์มาใช้ก็จะใช้วิธีการคล้ายกันในตาราง ดังนั้นผู้อ่านจะสามารถนำโปรแกรมนี้ไปปรับปรุงเพิ่มเติมต่อได้เองโดยไม่ยาก

ครานี้ก็มาว่ากันถึงเรื่องฟอร์มหลักที่ต้องออกแบบก่อน ส่วนประกอบต่าง ๆ ของฟอร์มผู้เขียนแสดงไว้แล้วในตารางที่ 1 ในฟอร์มหลักของโปรแกรมนี้ผู้เขียนกำหนดให้ขอบของหน้าต่างเป็นแบบคงที่เส้นเดียว และกำหนดหัวหน้าต่างที่พรอพเพอร์ตี้ส์ Caption ใหม่เป็น "Database Explorer" เราจะกำหนดค่าคอนโทรลเลเบล label1 เป็น "Tables:" และใช้คอนโทรลลิสต์บ็อกซ์ lstTbName สำหรับแสดงรายชื่อตารางทั้งหมดที่เก็บอยู่ในดาต้าเบสที่เปิดขึ้นมา จากรูปที่ 1 เป็นตัวอย่างเมื่อทดลองใช้ Database Explorer เปิดไฟล์ BIBLIO.MDB โปรแกรมจะแสดงรายชื่อตารางทั้งหมดในลิสต์บ็อกซ์ เพื่อรอให้ผู้ใช้เลือกตารางที่ต้องการขึ้นมาดู

การแสดงทั้งโครงสร้างตารางและข้อมูลของตารางจะใช้คอนโทรลกริดช่วย โดยกริดที่ใช้แสดงโครางสร้างเราจะตั้งชื่อเป็น grdStruct และกริดแสดงข้อมูลคือ grdRecord พรอพเพอร์ตี้ส์ HighLight ของกริดทั้งสองนี้กำหนดเป็น False พรอพเพอร์ตี้ส์ FixedCols และ FixedRows กำหนดเป็น 0 สำหรับ grdStruct กำหนดค่าเริ่มต้นของพรอพเพอร์ตี้ส์ Cols เป็น 3 เพราะเราจะใช้กริดแสดงชื่อ ขนาด และชนิดของฟิลด์ในตารางที่เลือก (ดูรูปที่ 2)



รูปที่ 1



รูปที่ 2

ออปเจ็กต์	พรอพเพอร์ตี้ส์	ค่าเริ่มต้น
VB.Form	Name	Form1
	BorderStyle	1 'Fixed Single
	Caption	"Database Explorer"
VB.ListBox	Name	lstTbName
	Visible	0 'False
VB.Line	Name	Line1
	BorderColor	&H00FFFFFF& 'White
MSGrid.Grid	Name	grdRecord
	Visible	0 'False
	FixedRows	0
	FixedCols	0
	HighLight	0 'False
MSGrid.Grid	Name	grdStruct
	Visible	0 'False
	Cols	3
	FixedRows	0
	FixedCols	0
	HighLight	0 'False
MSComDlg.CommonDialog	Name	cmdlg
VB.Label	Name	Label1
	BorderStyle	1 'Fixed Single
	Caption	"Tables: "
	Visible	0 'False
VB.Menu	Name	mnuFile
	Caption	"&File"
VB.Menu	Name	mnuOpenDb
	Caption	"&Open Database"
VB.Menu	Name	mnuOpenTb
	Caption	"Open &Table"
	Enabled	0 'False
	Visible	0 'False
VB.Menu	Name	mnuCloseTb
	Caption	"Close T&able"
	Enabled	0 'False
	Visible	0 'False
VB.Menu	Name	mnuSep0
	Caption	"-"
	Visible	0 'False
VB.Menu	Name	mnuCloseDb
	Caption	"&Close Database"
	Enabled	0 'False
VB.Menu	Name	mnuSep1
	Caption	"-"
VB.Menu	Name	mnuExit
	Caption	"E&xit"
VB.Menu	Name	mnuView
	Caption	"&View"
	Visible	0 'False
VB.Menu	Name	mnuViewRec
	Caption	"&Record"
	Enabled	0 'False
VB.Menu	Name	mnuViewStru
	Caption	"&Structure"
	Enabled	0 'False

ตารางที่ 1 แสดงออปเจ็กต์ต่าง ๆ ที่ต้องสร้างและค่าเริ่มต้นของแต่ละออปเจ็กต์

เมนูที่ออกแบบโปรแกรมนี้มีเมนูหลัก 2 เมนู คือ File และ View แต่เราจะกำหนดพรอพเพอร์ตี้ส์ Visible ให้เป็น False เพื่อให้ตอนโปรแกรมเริ่มทำงานจะยังไม่เห็นเมนู View สำหรับเมนูย่อยของ File จะประกอบด้วย Open Database, Open Table, Close Table, Close Database และ Exit จะเห็นว่าเรากำหนดให้ตอนเริ่มต้นพรอพเพอร์ตี้ส์ Enabled และ Visible ของเมนู Open Table, Close Table จะกำหนดเป็น False เพื่อให้เริ่มต้นจะไม่สามารถเห็นและใช้เมนูดังกล่าวได้ สำหรับเมนู Close Database กำหนดเฉพาะพรอพเพอร์ตี้ส์ Enabled เท่านั้น เมนูย่อยของ View ประกอบด้วย Record และ Structure โดยกำหนดเฉพาะพรอพเพอร์ตี้ส์ Enabled เป็น False สำหรับพรอพเพอร์ตี้ส์ Enabled และ Visible ของเมนูเหล่านี้จะถูกกำหนดค่าใหม่ให้เหมาะสมระหว่างที่โปรแกรมทำงาน

โค้ดที่ต้องเขียน ในโปรแกรม Database Explorer

เริ่มกันที่โปรแกรมส่วนที่ 1 ซึ่งเราต้องเขียนในส่วน Declarations โค้ดในส่วนนี้เป็นค่าคงที่และตัวแปรต่าง ๆ ที่ใช้ในโปรแกรม ขณะที่โปรแกรมเริ่มทำงาน (ดูรูปที่ 1) เราจะยังไม่เห็นคอนโทรล label1 และ lstTbName ในซับรูทีน Form_Load (โปรแกรมส่วนที่ 2) เริ่มด้วยการกำหนดค่าตัวแปร txtHead และ typeData ซึ่งเป็นสตริงที่ใช้แสดงในตารางโครงสร้างข้อมูล และกำหนดตัวแปร dbOpenFlg ที่เก็บสถานะว่าดาต้าเบสเปิดอยู่หรือไม่ และสุดท้ายก็จัดฟอร์มให้แสดงอยู่กลางจอภาพ

หากผู้อ่านลองเลือกเมนู File (ดูรูปที่ 3) จะเห็นว่าสามารถเลือกได้เฉพาะ Open Database (ดูรูปที่ 4) ส่วนเมนู Close Database ยังใช้ไม่ได้เพราะพรอพเพอร์ตี้ส์ Enabled มีค่าเริ่มต้นเป็น False จึงเท่ากับเป็นการบังคับให้ผู้ใช้ต้องเปิดดาต้าเบสเท่านั้น โค้ดที่ต้องเขียนในเมนูนี้แสดงในโปรแกรมส่วนที่ 3 โดยเราจะเลือกไฟล์ไดอะล็อกบ็อกซ์ของคอนโทรล CommonDialog ซึ่งโปรแกรมนี้ถูกเขียนบนวินโดวส์ 95 ดังนั้นไดอะล็อกบ็อกซ์จึงเป็นแบบ 32 บิต ซึ่งสนับสนุนการใช้ชื่อไฟล์แบบยาวด้วย

```
Option Explicit

Const MB_OK = 0
Const MB_YESNO = 4
Const MB_ICONQUESTION = 32
Const MB_ICONEXCLAM = 48

Dim myDb As Database
Dim myTable As Table
Dim myTableDef As TableDef

Dim dbFileName As String
Dim dbName As String
Dim tbName As String
Dim dbTableNums As Integer
Dim dbOpenFlg As Integer

Dim txtHead(4) As String
Dim typeData(12) As String
```

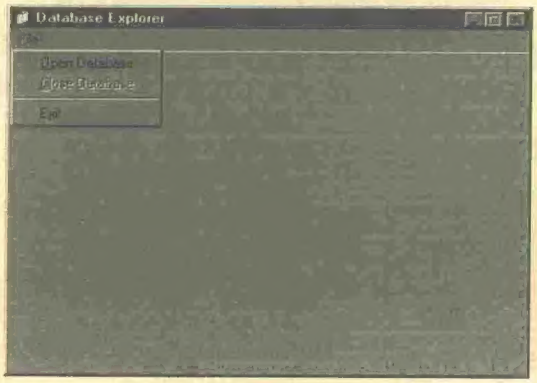
โปรแกรมส่วนที่ 1

โปรแกรมส่วนที่ 2

```
Private Sub Form_Load()
    InitTxtHead
    InitTypeData
    dbOpenFlg = False
    Top = (Screen.Height - Height) \ 2
    Left = (Screen.Width - Width) \ 2
End Sub

Private Sub InitTxtHead()
    txtHead(0) = "Name      "
    txtHead(1) = "Type      "
    txtHead(2) = "Size      "
End Sub

Private Sub InitTypeData()
    typeData(0) = "Boolean"
    typeData(1) = "Byte"
    typeData(2) = "Integer"
    typeData(3) = "Long"
    typeData(4) = "Currency"
    typeData(5) = "Single"
    typeData(6) = "Double"
    typeData(7) = "Date"
    typeData(8) = "Binary"
    typeData(9) = "Text"
    typeData(10) = "Long Binary"
    typeData(11) = "Memo"
End Sub
```

รูปที่ 3

รูปที่ 4



สำหรับชื่อไฟล์ที่ผู้ใช้เลือกจะคืนกลับมาที่พรอพเพอร์ตี้ส์ Filename ของคอนโทรล cmdlg ซึ่งเราจะเก็บไว้ที่ตัวแปร dbFileName และเก็บเฉพาะชื่อไฟล์อย่างเดียวไม่รวมพาธไว้ที่ตัวแปร dbName จากนั้นจึงเรียกซับรูทีน OpenMyDb ให้เปิดไฟล์ที่เลือก เมื่อเปิดดาต้าเบสสำเร็จก็จะกำหนดสถานะตัวแปร dbOpenFlg ให้เป็น True

ซับรูทีน OpenMyDb จะเริ่มด้วยการกำหนดพรอพเพอร์ตี้ส์ Visible ของคอนโทรล label1, lstTbName, เมนู Open Table, Close Table, View เป็น True เพื่อให้ผู้ใช้เห็นเมนูและคอนโทรลเหล่านี้ (ดูรูปที่ 1) กำหนดพรอพเพอร์ตี้ส์ Enabled ของเมนู Close Database ให้เป็น True เพื่อใช้ปิดดาต้าเบสได้ และซ่อนเมนู Open Database โดยพรอพเพอร์ตี้ส์ Visible ให้เป็น False เปลี่ยนชื่อหัวหน้าต่างให้แสดงชื่อดาต้าเบสที่เปิด ต่อจากนั้นก็ต้องเปิดดาตาเบสด้วยฟังก์ชัน OpenDatabase และใช้ประโยค Set กำหนดค่าออปเจ็กต์ myDb ต่อไปก็ถึงขั้นค้นหาชื่อตารางทั้งหมดที่มีในดาต้าเบสเพื่อนำมาแสดงในลิสต์บ็อกซ์ lstTbName โดยเลือกเฉพาะชื่อตารางที่ไม่ได้ขึ้นต้นด้วย "Msys" ขึ้นมาใช้ จำนวนตารางของดาต้าเบสจะเก็บอยู่ในตัวแปร dbTableNums

```
Private Sub mnuOpenDb_Click()
    Const OFN_HIDEREADONLY = &H4

      On Error GoTo openErr
      cmdlg.Flags = OFN_HIDEREADONLY
      cmdlg.DialogTitle = "Open Database"
      cmdlg.Filter = "Access Database(*.mdb)|*.mdb"
      cmdlg.Action = 1
      dbFileName = cmdlg.filename
      dbName = UCase$(cmdlg.FileTitle)
      OpenMyDb
      dbOpenFlg = True
openExit:
    Exit Sub
openErr:
      MsgBox Error$(Err), MB_OK + MB_ICONEXCLAM,
      "Database Explorer"
      GoTo openExit
End Sub

Private Sub OpenMyDb()
      Dim i As Integer

      On Error GoTo openDbError

      lstTbName.Visible = True
      label1.Visible = True
      mnuCloseDb.Enabled = True
      mnuOpenDb.Visible = False
      mnuOpenTb.Visible = True
      mnuCloseTb.Visible = True
      mnuSep0.Visible = True
      mnuView.Visible = True
      Caption = dbName

      Set myDb = OpenDatabase(dbFileName)

      dbTableNums = 0
      For i = 0 To myDb.TableDefs.Count - 1
        If Left$(myDb.TableDefs(i).Name, 4)
      <> "MSys" Then
          dbTableNums = dbTableNums + 1
         lstTbName.AddItem myDb.TableDefs(i).Name
        End If
      Next
openDbExit:
      Exit Sub
openDbError:
      MsgBox Error$(Err), MB_OK + MB_ICONEXCLAM,
      "Database Explorer"
      GoTo openDbExit
End Sub
```

โปรแกรมส่วนที่ 3

หลังจากเปิดดาต้าเบสแล้วถ้าผู้ใช้เลือกเมนู File ดูก็จะเห็นว่า เมนู Open Table และ Close Table มีอยู่ยังใช้ไม่ได้ เนื่องจากยังดิสเอเบิลอยู่ ผู้ใช้ต้องเลือกชื่อตารางที่จะเปิดโดยคลิ๊กในลิสต์บ็อกซ์ ซึ่งซับรูทีน lstTbName_Click จะถูกเรียกมาทำงาน เราต้องเขียนโค้ดในส่วนโปรแกรมที่ 4 โดยกำหนดพรอพเพอร์ตี้ส์ Enabled ของเมนู Open Table ให้เป็น True เพื่อให้ผู้ใช้สั่งเปิดตารางที่เลือกได้ ส่วนพรอพเพอร์ตี้ส์ Enabled ของเมนู Close Table ยังมีค่าเป็น False อยู่ ชื่อตารางที่ผู้ใช้เลือกจะเก็บไว้ในตัวแปร tbName

โค้ดส่วนถัดไปจะเขียนสำหรับเมนู Open Table (ดูโปรแกรมส่วนที่ 5) เริ่มด้วยการเซตพรอพเพอร์ตี้ส์ Enabled ของเมนู Close Table ให้เป็น True ส่วนเมนู Open Table เซตเป็น False เพื่อไม่ให้เปิดตารางซ้ำ จากนั้นจึงกำหนดค่าออปเจ็กต์ myTable ด้วยเมธอด OpenTable และออปเจ็กต์ myTableDef โดยใช้พรอพเพอร์ตี้ส์ TableDefs Collection เมื่อเปิดตารางเรียบร้อยแล้วก็จะกำหนดพรอพเพอร์ตี้ส์ Enabled ของเมนู Record และ Structure ให้เป็น True เพื่อให้ผู้ใช้เลือกได้ว่าจะเลือกดูโครงสร้างหรือข้อมูลของตาราง จากนั้นก็แก้ที่ชื่อหัวหน้าต่างโดยเพิ่มชื่อตารางที่เลือกเข้าไปด้วย แล้วจึงเรียกซับรูทีน GetStruc และ GetRecord เพื่อนำรายละเอียดโครงสร้างตารางใส่ในกริด grdStruct และนำเรคอร์ดของตารางมาใส่ในกริด grdRecord

ซับรูทีน GetStruc มีหน้าที่เตรียมรายละเอียดโครงสร้างของตารางมาใส่ไว้ในกริด grdStruct โดยกำหนดจำนวนแถวของกริดให้เท่ากับจำนวนฟิลด์ของตารางบวกหนึ่ง (รวมหัวตาราง) เริ่มจากแถวที่หนึ่ง (เซตค่า Row เป็น 0) ที่เราใช้แสดงหัวตาราง โดยนำค่ามาจาก txtHead มากำหนดให้พรอพเพอร์ตี้ส์ Text ของกริด และต้องปรับความกว้างของคอลัมน์กริดให้กว้างเป็น 1.5 เท่าของความกว้างของสตริงที่แสดงในคอลัมน์ จากนั้นก็เฉลี่ยความกว้างของคอลัมน์ทั้งสามให้เท่ากันอีกครั้ง เมื่อสร้างหัวตารางเสร็จแล้ว ก็จะเริ่มต้นอ่านโครงสร้างของตารางทีละฟิลด์มาใส่ในกริด โดยในคอลัมน์แรกจะเป็นชื่อฟิลด์ คอลัมน์ที่สองเป็นชนิดข้อมูลซึ่งเราต้องนำตัวเลขที่อ่านได้ไปดึงสตริงจากอะเรย์ typeData มาใช้ คอลัมน์ที่สามจะอ่านค่าขนาดของข้อมูลที่เก็บในฟิลด์

ซับรูทีน GetRecord (ดูที่โปรแกรมส่วนที่ 7) มีหน้าที่อ่านเรคอร์ดข้อมูลในตารางทั้งหมดขึ้นมาใส่ไว้ในกริด grdRecord (ดูรูปที่ 5) เริ่มจากกำหนดจำนวนคอลัมน์ของกริดด้วยจำนวนฟิลด์ของตารางและกำหนดแถวของกริดให้มีแค่ 2 แถวก่อนเพราะเรายังไม่รู้ว่าข้อมูลทั้งหมดมีกี่เรคอร์ด จากนั้นก็นำชื่อฟิลด์มาใส่ไว้ที่แถวแรกในกริด และอ่านค่าของข้อมูลแต่ละฟิลด์ขึ้นมาเตรียมเป็นเท็กซ์เก็บไว้ใน rowTxt โดยแต่ละฟิลด์จะแยกกันด้วยแท็บหรือ Chr$(9) จากนั้นก็ใช้เมธอด AddItem เพิ่มแถวข้อมูลแต่ละเรคอร์ดเข้าไปในกริด ข้อมูลทั้งหมดในตารางจะถูกอ่านขึ้นมาและตรวจสอบว่าหมดตารางเมื่อค่าพรอพเพอร์ตี้ส์ EOF เป็น True เมื่อใส่ค่าทั้งหมดในกริดแล้ว ก็จะต้องปรับความกว้างของกริดแต่ละคอลัมน์อีกครั้ง

โปรแกรมส่วนที่ 4

```
Private Sub lstTbName_Click()
    mnuOpenTb.Enabled = True
    tbName = lstTbName.Text
End Sub
```

โปรแกรมส่วนที่ 5

```
Private Sub mnuOpenTb_Click()
    mnuCloseTb.Enabled = True
    mnuOpenTb.Enabled = False

    Set myTable = myDb.OpenTable(tbName)
    Set myTableDef = myDb.TableDefs(tbName)

    mnuViewStru.Enabled = True
    mnuViewRec.Enabled = True
    Caption = Caption + " : " + tbName
    GetStruc
    GetRecord
End Sub
```

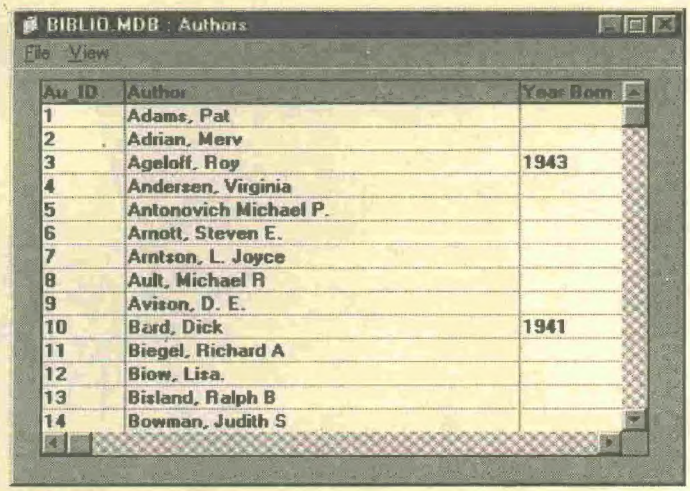
โปรแกรมส่วนที่ 6

```
Private Sub GetStruc()
    Dim i As Integer
    Dim ColWidth As Integer
    Dim totalWidth As Integer

    grdStruct.Rows =
myTableDef.Fields.Count + 1
    grdStruct.Row = 0
    For i = 0 To 2
        grdStruct.Col = i
        grdStruct.Text = txtHead(i)
        ColWidth = 1.5 *
TextWidth(txtHead(i))
        grdStruct.ColWidth(i) = ColWidth
* Screen.TwipsPerPixelX
        totalWidth = totalWidth +
ColWidth
    Next
    ColWidth = (grdStruct.Width -
totalWidth) * Screen.TwipsPerPixelX / 3
    If (ColWidth > 0) Then
       For i = 0 To 2
          grdStruct.ColWidth(i) =
ColWidth + grdStruct.ColWidth(i)
       Next
    End If
    grdStruct.FixedRows = 1
    For i = 1 To grdStruct.Rows - 1
        grdStruct.Row = i
        grdStruct.Col = 0
        grdStruct.Text =
myTableDef.Fields(i - 1).Name
        grdStruct.Col = 1
        grdStruct.Text =
typeData(myTableDef.Fields(i - 1).Type -
1)
        grdStruct.Col = 2
        grdStruct.Text =
myTableDef.Fields(i - 1).Size
    Next
End Sub
```

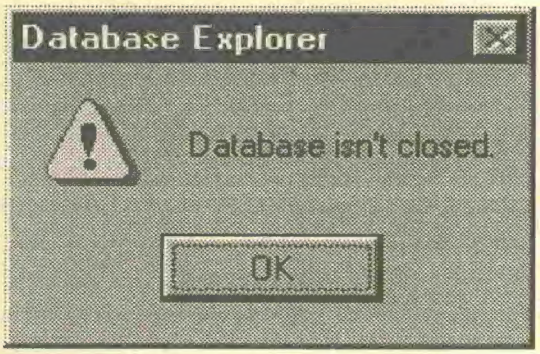
โปรแกรมส่วนที่ 7

```
Private Sub GetRecord()
    Dim i As Integer
    Dim ColWidth() As Integer
    Dim txtWidth As Integer
    Dim grdCols As Integer
    Dim colTxt As String
    Dim rowTxt As String
    Dim rowIst As Integer

    On Error GoTo getRecError

    grdCols = myTableDef.Fields.Count
    grdRecord.Rows = 2
    grdRecord.Cols = grdCols

    ReDim ColWidth(grdCols) As Integer

    grdRecord.Row = 0
    For i = 0 To grdCols - 1
        colTxt = myTableDef.Fields(i).Name
        grdRecord.Col = i
        grdRecord.Text = colTxt
        ColWidth(i) = 1.5 * TextWidth(colTxt)
    Next
    grdRecord.FixedRows = 1
    rowIst = True
    Do While Not myTable.EOF
       rowTxt = ""
       For i = 0 To grdCols - 1
           If Not IsNull(myTable.Fields(i).Value) Then
              colTxt = myTable.Fields(i).Value
           Else
              colTxt = " "
           End If
           rowTxt = rowTxt & colTxt & Chr$(9)
           txtWidth = 1.5 * TextWidth(colTxt)
           If txtWidth > ColWidth(i) Then
              ColWidth(i) = txtWidth
           End If
       Next
       If rowIst Then
          grdRecord.SelStartRow = 1
          grdRecord.SelEndRow = 1
          grdRecord.SelStartCol = 0
          grdRecord.SelEndCol = grdCols - 1
          grdRecord.Clip = rowTxt
          rowIst = False
       Else
          grdRecord.AddItem Left$(rowTxt, Len(rowTxt) - 1)
       End If
       myTable.MoveNext
    Loop
    For i = 0 To grdCols - 1
        grdRecord.ColWidth(i) = ColWidth(i) * Screen.TwipsPerPixelX
    Next
getRecExit:
    Exit Sub
getRecError:
    MsgBox Error$(Err), MB_OK + MB_ICONEXCLAM, "Database Explorer"
    GoTo getRecExit
End Sub
```

หลังจากที่เปิดตารางแล้วผู้ใช้จะทำงานต่อได้ทางเดียวคือ ต้องเลือกเมนู Close Table เท่านั้น เพราะเมนูอื่น ๆ ดิสเอเบิลหมด ดังนั้นเราก็มาดูโค้ดที่จะเขียนสำหรับเมนูนี้ซึ่งแสดงอยู่ในส่วนที่ 8 เริ่มจากเราต้องยกเลิกชื่อตารางที่ผู้ใช้เลือกไว้ก่อนหน้านี้ซึ่งเราทราบตำแหน่งจากค่า lstTbName.ListIndex โดยกำหนดพรอพเพอร์ตี้ส์ Selected ของลิสต์ตำแหน่งนั้นให้เป็น False จากนั้นก็กำหนดพรอพเพอร์ตี้ส์ Enabled ของเมนู Open Table, Close Table, Record, Structure ให้เป็น False ส่วนของเมนู Close Database กำหนดเป็น True ต่อไปก็ปิดตารางด้วยเมธอด Close ซ่อนกริดทั้งสองอันด้วยการกำหนดพรอพเพอร์ตี้ส์ Visible เป็น False จากนั้นนำคอนโทรล label1 และ lstTbName กลับมาแสดงใหม่ แก้ไข



รูปที่ 5



รูปที่ 6

ค่า Caption ของฟอร์มให้แสดงเฉพาะชื่อดาต้าเบส ดังนั้นเมื่อถึงจุดนี้ผู้ใช้มีทางเลือกเพียง 2 ทาง คือปิดดาต้าเบสที่ใช้อยู่ หรือกลับไปเลือกชื่อตารางอื่นที่แสดงอยู่ลิสต์บ็อกซ์เพื่อเปิดตารางอื่นมาดูต่อ

เมื่อผู้ใช้เลือกเมนู Close Database เพื่อปิดดาต้าเบส โปรแกรมจะเรียกซับรูทีน CloseMyDb ให้ปิดดาต้าเบสด้วยเมธอด Close (ดูโปรแกรมส่วนที่ 9) ซ่อนคอนโทรล label1 และ lstTbName แก้ไข Caption ของฟอร์มเป็น Database Explorer จากนั้นก็ซ่อนเมนู Open Table, Close Table, View และกริดทั้งสองอัน ลบข้อมูลทั้งหมดในลิสต์บ็อกซ์ด้วยเมธอด RemoveItem และเซตสถานะของ dbOpenFlg ให้เป็น False

โค้ดของเมนู Record และ Structure นั้นจะมีหน้าที่เพียงกำหนดพรอพเพอร์ตี้ส์ Visible ของกริดแต่ละกริดให้แสดงถูกต้องในแต่ละกรณี นอกจากนั้นก็ต้องซ่อน label1, lstTbName และคิสเอเบิลเมนู Close Database เท่านั้น

เมนู Exit นั้นเราจะคอยดักกรณีที่ผู้ใช้ไม่ได้ปิดดาต้าเบสไว้ใน Form_Unload โดยถ้าดาต้าเบสยังไม่ปิดค่า dbOpenFlg จะยังเป็น True โปรแกรมจะแสดงไดอะล็อกบ็อกซ์แจ้งให้ผู้ใช้ทราบว่ายังไม่ได้ปิดไฟล์ (ดูรูปที่ 6) และเซตค่า Cancel ให้เป็น True เพื่อยกเลิกเหตุการณ์ Unload กรณีที่ดาต้าเบสปิดแล้วโปรแกรมก็จะจบการทำงานด้วยคำสั่ง End

```
Private Sub mnuCloseTb_Click()
      On Error GoTo closeTbError
      lstTbName.Selected(lstTbName.ListIndex)
                                 = False
      mnuOpenTb.Enabled          = False
      mnuCloseTb.Enabled         = False
      mnuViewRec.Enabled         = False
      mnuViewStru.Enabled        = False
      mnuCloseDb.Enabled         = True

      myTable.Close

      grdStruct.Visible = False
      grdRecord.Visible = False
      label1.Visible = True
      lstTbName.Visible = True
      Caption = dbName
closeTbExit:
      Exit Sub
closeTbError:
      MsgBox Error$(Err), MB_OK + MB_ICONEXCLAM,
      "Database Explorer"
      GoTo closeTbExit
End Sub
```

โปรแกรมส่วนที่ 8

```
Private Sub mnuCloseDb_Click()
      On Error GoTo closeErr
      CloseMyDb
closeExit:
      Exit Sub
closeErr:
      MsgBox Error$(Err), MB_OK + MB_ICONEXCLAM,
      "Database Explorer"
      GoTo closeExit
End Sub

Private Sub CloseMyDb()
      Dim i As Integer

      On Error GoTo closeDbError

      myDb.Close

      lstTbName.Visible = False
      label1.Visible = False
      Caption = "Database Explorer"
      mnuCloseDb.Enabled = False
      mnuOpenDb.Visible = True
      mnuOpenTb.Visible = False
      mnuCloseTb.Visible = False
      mnuSep0.Visible = False
      mnuView.Visible = False
      grdStruct.Visible = False
      grdRecord.Visible = False
      For i = 0 To lstTbName.ListCount - 1
        lstTbName.RemoveItem 0
      Next
      dbOpenFlg = False
closeDbExit:
      Exit Sub
closeDbError:
      MsgBox Error$(Err), MB_OK + MB_ICONEXCLAM,
      "Database Explorer"
      GoTo closeDbExit
End Sub
```

โปรแกรมส่วนที่ 9

```
Private Sub mnuViewRec_Click()
      If grdStruct.Visible Then
       grdStruct.Visible = False
      End If
      label1.Visible = False
      lstTbName.Visible = False
      grdRecord.Visible = True
      mnuCloseDb.Enabled = False
End Sub

Private Sub mnuViewStru_Click()
      If grdRecord.Visible Then
       grdRecord.Visible = False
      End If
      label1.Visible = False
      lstTbName.Visible = False
      grdStruct.Visible = True
      mnuCloseDb.Enabled = False
End Sub
```

โปรแกรมส่วนที่ 10

```
Private Sub Form_Unload(Cancel As Integer)
      If dbOpenFlg Then
      MsgBox "Database isn't closed.",
       MB_OK+MB_ICONEXCLAM, "Database Explorer"
       Cancel = True
      Else
       End
      End If
End Sub

Private Sub mnuExit_Click()
      Form_Unload False
End Sub
```

โปรแกรมส่วนที่ 11

มองต่างมุม

บทความตอนนี้ได้นำเสนอมุมมองของดาต้าออปเจ็กต์ในแง่มุมใหม่ ซึ่งคิดว่าน่าจะทำให้เห็นภาพของดาต้าออปเจ็กต์ได้ดีขึ้น และสามารถนำไปประยุกต์ใช้งานต่อได้เป็นอย่างดี ความจริงยังมีเรื่องราวของการใช้ดาต้าออปเจ็กต์อีกหลายอย่าง หากมีโอกาสก็คงจะได้นำมาฝากกันใหม่

รับสมัครงาน

บริษัท คอมมิวนิเคชั่น อีเลคทรอนิคส์ ซัพพลาย จำกัด
เป็นผู้จัดจำหน่ายสินค้าอีเลคทรอนิคส์ ขณะนี้กำลังขยายงาน ต้องการผู้ร่วมงานหลายตำแหน่งดังต่อไปนี้

1. ช่างอีเลคทรอนิคส์ 5 ตำแหน่ง
- ชาย/สารพัดช่าง, ช่างติดตั้งงานไฟฟ้า
- ปวช.

2. ผู้ช่วยการตลาด 3 ตำแหน่ง
- ชาย/ม.6 ขึ้นไป
- มีใบขับขี่รถมอเตอร์ไซค์

3. พนักงานคอมพิวเตอร์ 2 ตำแหน่ง
- ชาย/ปวช.,ปวส.

4. พนักงานจัดส่ง 3 ตำแหน่ง
- ชาย/วุฒิ ป.6 ขึ้นไป
- มีใบขับขี่รถมอเตอร์ไซค์

สนใจโปรดสมัครด้วยตนเองหรือส่งจดหมายมาพร้อมหลักฐานการศึกษามาที่ฝ่ายบุคคล

บริษัท คอมมิวนิเคชั่น อีเลคทรอนิคส์ ซัพพลาย จำกัด
234/2 ซ.ประชาชื่น 37 (นพเก้า 5) บางซื่อ กรุงเทพฯ 10800
โทรศัพท์ (02) 910-0411-5

บริษัทมีความประสงค์ที่จะรับสมัครเจ้าหน้าที่เข้าประจำตำแหน่งต่าง ๆ ดังนี้

1. พนักงานขาย

2. พนักงานธุรการ

3. พนักงานบัญชี
- วุฒิการศึกษา ปวช. ขึ้นไป
- มีมนุษยสัมพันธ์ดี
- ถ้ามีประสบการณ์จะได้รับการพิจารณาเป็นพิเศษ

ผู้ที่สนใจสมัครด้วยตนเองหรือส่งจดหมายพร้อมหลักฐานมาสมัครได้ที่

บริษัท กาญจนกิตต์ มาร์เก็ตติ้ง จำกัด
18/7 หมู่ 9 ถ.เพชรเกษม 50-52 แขวงบางหว้า เขตภาษีเจริญ กรุงเทพฯ 10160
TEL. 455-8612, 454-3649, 869-6338-9